

Simulador Interativo do Oscilador de Duffing: Um Estudo de Dinâmica Não-Linear e
Caos

Monografia do trabalho da disciplina de Oficina de Integração I sob o título “Simulador
Interativo do Oscilador de Duffing: Um Estudo de Dinâmica Não-Linear e Caos”,
aprovada por:

Prof Dr Mário Sérgio Teixeira de Freitas

Licença

Este trabalho está licenciado sob uma Licença Creative Commons - Atribuição - Compartilhamento pela mesma Licença 2.5 Brasil. Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/br/> ou envie uma carta para Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Agradecimentos

Agradecemos aos professores que ministram a disciplina de Oficina de Integração I: Arandi Ginane Bezerra Jr., Luiz Ernesto Merkle, Gustavo Alberto Gimenez-Lugo que foram os incentivadores à realização desse trabalho e nos proporcionaram um processo contínuo de aprendizado. E ao orientador Mário Sérgio Teixeira de Freitas, que contribuiu enormemente para a elaboração deste trabalho e nos despertou interesse pela área de Dinâmica não-linear e Caos.

Resumo

A seguinte monografia tem como objetivo o desenvolvimento e documentação de um aplicativo que auxilie no aprendizado sobre dinâmica não-linear e caos. O aplicativo tem como objeto de estudo o modelo do oscilador não-linear de Duffing. Seu comportamento caótico proporciona a implementação de diversas ferramentas interessantes ao estudo da não-linearidade tais como a simulação gráfica de seu comportamento físico dentre outras. O aplicativo utiliza-se de métodos numéricos para relacionar-se ao modelo físico do Oscilador de Duffing. Para o desenvolvimento deste trabalho foi necessário o estudo teórico no que concernem as diversas disciplinas envolvidas (Física, Matemática, Estrutura de Dados, Engenharia de Software, dentre outras) e configurar os recursos e aplicações que o software, anteriormente em projeto, possui.

Palavras-chave: Caos e Dinâmica não-linear, Oscilador de Duffing, Double-well, Simulador.

Abstract

The present monograph aims to develop and substantiate a software that helps on the learning of chaos and non-linear dynamics. The application has model called Duffing's non-linear oscillator as object of study. Its chaotic behavior allows the implement of several tools that are interesting for the study of non-linearity such as the graphical simulation of its physical behavior and others. The application rely on the use of numerical methods to link itself to the physical model of Duffing's oscillator. To accomplish this paper it was required the theoretical study that involves the several linked matters (such as Physics, Mathematics, Data Structure, Software Engineering and others) and the setting of features and applications that the software, previously in project, has.

Key-words: Non-linear dynamics and Chaos, Duffing's oscillator, Double-well, simulator.

Sumário

Licença	4
Agradecimentos	5
Resumo	6
Abstract	7
1 INTRODUÇÃO	10
1.1 Motivação	10
1.2 Justificativa	10
1.3 Objetivos	12
1.3.1 Gerais	12
1.3.2 Específicos	12
1.4 Metodologia	12
1.5 Estrutura do Trabalho	12
2 ENSINO	13
2.1 Ensino de Física	14
2.1.1 Simulações Computacionais no Ensino de Física	14
2.2 Ensino de Dinâmica Não-Linear e Caos	16
2.2.1 A utilização do software na disciplina de Dinâmica não-linear e Caos	16
3 Dinâmica Não-Linear e Caos	17
3.1 Oscilador de Duffing	20
3.1.1 Gráficos	21
4 Software	23
4.0.2 Engenharia de Software	24
4.1 Análise de Aplicativos Similares	25
4.2 Padrões de Projeto	27
4.2.1 Padrão MVC	27
4.2.2 Importância do Padrão de Projeto Observer no Sistema	28
4.3 UML	29

4.4	JAVA	29
4.5	Interface Gráfica	29
4.5.1	Ícones	30
4.6	Análise Orientada a Objetos	30
4.6.1	Descrição do Problema	30
4.6.2	Requisitos Funcionais do Sistema	31
4.6.3	Diagrama de Classes	32
4.6.4	Diagramas de Sequencia	33
4.6.5	Dicionário de Classes	35
5	Considerações Finais	37
5.1	Discussão	37
5.2	Resultados	38
5.3	Conclusão	40
6	Referências Bibliográficas	41
7	Apêndice	44
7.1	Um breve manual de utilização	44
7.1.1	Janelas Internas	44
7.1.2	Abrindo Janelas Internas	46
7.1.3	Iniciando Simulação	46

1 INTRODUÇÃO

1.1 Motivação

1.2 Justificativa

A disciplina de Oficina de Integração I do curso de Engenharia de Computação ofertado na UTFPR tem como ementa o seguinte texto:

“Integração dos conhecimentos de disciplinas de formação básica e profissionalizante obtidos até o momento. Aplicação dos conceitos de metodologia científica para o desenvolvimento, em equipes, de um sistema computacional (software) contemplando essa integração. Aplicação dos conceitos de metodologia científica e comunicação e expressão para a elaboração e apresentação oral de relatório final dos resultados do projeto desenvolvido.” (UTFPR, 2006)

Ao analisar a ementa pode-se observar que Oficina de Integração I tem como objetivos a multidisciplinariedade e a integração de outras disciplinas. Essa visão é reforçada no artigo “Interdisciplinariedade e Criticidade na Formação em Engenharia de Computação na UTFPR” pois, segundo os autores, “...as disciplinas de oficinas de integração são, ao mesmo tempo, estratégia e ferramenta para promover a integração curricular”.(MERKLE; BEZERRA; GIMENÉZ-LUGO, 2008).

Tendo em vista este contexto, o trabalho desenvolvido se encaixa nos requisitos da disciplina. Pois o desenvolvimento de um aplicativo que simule o modelo do oscilador não-linear de Duffing exige a integração de várias disciplinas tais como: Física, Dinâmica não-linear, Cálculo Numérico, Fundamentos de Programação e Engenharia de Software.

Além de atender os objetivos da disciplina, outra grande motivação para a execução deste projeto foi a falta de um aplicativo que reúna todas as ferramentas que o software desenvolvido neste trabalho possui e que são necessárias para um bom entendimento tanto de dinâmica não-linear caótica quanto do próprio oscilador de Duffing *****COLOCAR ALGUMA REFERENCIA DA ANÁLISE DE SIMILARES*****. É importante a diversificação das ferramentas de ensino de Física e Matemática. A modelagem computacional proporciona a interação entre os recursos áudio-visuais de forma simples e dinâmica, portanto, favorecendo o aprendizado. *****Alguns aplicativos **REFERENCIA À ANÁLISE** foram desenvolvidos com o mesmo intuito, porém não

possuem as ferramentas julgadas importantes no bom entendimento de dinâmica não-linear e caos. Isto fica evidente em um relato de experiências realizadas em sala de aula com o Modellus, um software de modelagem interativa com matemática, que não apresentou forte dependência das condições iniciais do oscilador de Duffing. (MARIO BOLDO; CIANI, 2007).

1.3 Objetivos

1.3.1 Gerais

Auxiliar o ensino de dinâmica não-linear e caos, no estudo de modelos não-lineares biestáveis caóticos, por meio de uma simulação computacional que o software desenvolvido proporciona. Tornando, assim, viáveis as simulações de experimentos custosos.

1.3.2 Específicos

Implementar algumas funcionalidades específicas que, de acordo com a visão deste projeto, são importantes para um bom aprendizado sobre dinâmica não-linear. São alguns exemplos dessas sub-rotinas: retrato ou plano de fase, seções temporais e uma animação que simule o comportamento físico do oscilador de Duffing que é chamada de espaço de configuração.

*** Revisão Bibliográfica sobre DNL, estudo de padrões de projeto, métodos numéricos ensino de física, etc. ***

1.4 Metodologia

1.5 Estrutura do Trabalho

2 ENSINO

O aprendizado é um processo ativo de construção ao invés da simples recepção ou aquisição de conhecimentos (FONTOURA,2002).

Com a difusão da computação, cada vez mais a educação tem como uma de suas ferramentas o computador. Porém, muitas vezes seu uso não é o mais adequado. Quando há computadores nas salas, muitas vezes eles são usados da mesma maneira que os outros objetos mais antigos como o caderno e o quadro. Alunos usam ?Personal Computers? somente para fazer anotações enquanto professores os usam apenas como usariam quadros negros digitais. Torres comenta o potencial que pode ter esse tipo de iniciativa:

?...O aspecto das ajudas técnico informáticas é enfatizado, tendo em vista que ele pode contribuir para uma acentuada melhoria, tanto no trabalho dos docentes como na aprendizagem dos alunos?(TORRES apud VIEIRA, 2002, p. 166).

Um dos pioneiros em usar ambientes virtuais para o ensino foi Seymour Papert, um educador matemático, que nos anos sessenta no MIT - Massachusetts Institute of Technology, de Cambridge, MA, Estados Unidos iniciou o projeto Logo. Adaptada para o português em 1982, na Unicamp, pelo Núcleo de Informática Aplicada à Educação (NIED), a linguagem ?Logo? vem sendo utilizada para trabalhar com crianças e adolescentes. ?Logo? é uma linguagem de programação, isto é, um meio de comunicação entre o computador e a pessoa que irá usá-lo. A principal diferença entre ?Logo? e outras linguagens de programação está no fato de que foi desenvolvida para ser usada por crianças e para que as crianças possam, com ela, aprender outras coisas.

Uma iniciativa interessante está no projeto Visual GD, tema de estudo da tese de doutorado de Josiane Wanderlinde Vieira, programa que tem como foco o ensino de geometria descritiva para alunos surdos. Através da hipermídia usou-se de vídeos, gráficos, texto e interatividade para ensinar um conteúdo relativamente abstrato para deficientes auditivos. **FAZER REFERENCIA AO PROJETO VISUAL GD**

Um dos grandes impecílios para o aumento do uso de aplicativos computacionais nas escolas é o conservadorismo, como afirma o professor da Unicamp Eduardo Chaves ?A tecnologia vem sendo introduzida na escola de forma conservadora, domesticada, o que apenas reforça o paradigma educacional tradicional, centrado na transmissão de informações disciplinares através do ensino do professor?(CHAVES, 2002). Porém essa

visão pode estar mudando. Pois ao reconhecer o computador como uma ferramenta interativa ?...representa criar mensagens para um novo paradigma de aprendizagem, no qual o aluno ao invés de ouvir e assimilar, interage ativamente com a tecnologia.? (ROMIS-ZOWSKI,2002) Isso entra em consenso com a afirmação de Viet, 2002 “...a modelagem computacional é uma ferramenta indispensável para o ensino de Física nos dias atuais”. Uma afirmação extremamente importante para este estudo, uma vez que o conteúdo do aplicativo a ser desenvolvido está inserido no campo da Física.

2.1 Ensino de Física

Ensino de Física tem se alterado significativamente nas últimas décadas devido a vários recursos computacionais que implementam novas ferramentas para o aprendizado (GLEISER, 2000).

A modelagem é de fundamental importância para o desenvolvimento do conhecimento educacional científico. Esta pode ser utilizada para melhorar o entendimento da Física. Uma das grandes críticas das estratégias didáticas é a desconexão que estas têm entre a prática e a teoria*****, de forma que a grande motivação no estudo da Física reside na relação existente entre situações cotidianas e os modelos conceituais. (BRANDÃO; ARAUJO; VEIT, 2008) O método científico consiste na elaboração de teorias que expliquem determinados comportamentos de fenômenos. A validade dessas teorias é verificada empiricamente. Todavia, na criação de modelos científicos há limitações e restrições referentes à aplicabilidade desses modelos. No ensino da Física, ao estabelecer certas idealizações e aproximações, podem-se relacionar situações do dia-dia com modelos físicos. Nesse contexto, a modelagem computacional permite a interação entre a abordagem teórica e a prática. (BRANDÃO; ARAUJO; VEIT, 2008)

?O computador, visto neste processo como uma ferramenta didática no auxílio da aprendizagem, pode fornecer oportunidades ímpares para a contextualização, visualização e apresentações das mais diversas situações físicas que possam dar sentido ao conceito físico que esteja sendo trabalhado pelo professor.? (BRANDÃO; ARAUJO; VEIT, 2008)

2.1.1 Simulações Computacionais no Ensino de Física

O ensino da Física envolve a abordagem de alguns conceitos abstratos, dos quais utiliza da Matemática para obter soluções. Além disso, a Física lida com objetos de

estudo fora do nível de percepção humana, como: partículas subatômicas, corpos dotados de grande velocidade e estudos complexos. Isto inviabiliza a representação de certos modelos físicos, pois os recursos em sala-de-aula são limitados. Outra dificuldade está na elaboração de representações estáticas de fenômenos dinâmicos. Alguns livros em busca de representar estes tipos de movimento utilizam até fotografias estroboscópicas. As animações computacionais resolvem o problema de representar tais modelos físicos pois um movimento, por exemplo, pode ser visualizado com extrema facilidade. As simulações computacionais oferecem maior interatividade entre os usuários e o computador. Elas possuem mecanismos que permitem ao usuário inserir valores de entrada, que são processados, e originam animações gráficas e outros recursos, como a elaboração de gráficos do movimento. (MEDEIROS,2002)

Os modelos computacionais são baseados em modelos físicos reais, que são expressos matematicamente, processados e reconstruídos através de animações. O principal uso da simulação computacional é quando o experimento real é inviável, seja por custos ou pela rapidez da realização, ou pela periculosidade do experimento ou pela interferência de fatores externos. (MEDEIROS,2002)

São apontados alguns benefícios da simulação computacional, como:

1. Reduzir o "ruído" do experimento para concentrar-se somente no fenômeno físico;
2. Fornecer um feedback para aperfeiçoar a compreensão dos conceitos físicos;
3. Permitirem aos estudantes coletarem grandes quantidades de dados em menor tempo;
4. Melhorar a compreensão dos fenômenos físicos;
5. Auxiliar os estudantes a aprenderem sobre o mundo natural, vendo e interagindo com os modelos científicos subjacentes que não poderiam ser inferidos através da observação direta;
6. Estabelecer as relações de causa e efeito de fenômenos complexos;
7. Aplicação de conceitos abstratos;

Entretanto, devem-se destacar algumas limitações impostas pelas simulações computacionais no ensino da Física. Como por exemplo, que um modelo real geralmente é complexo, e a representação deste em termos computacionais envolvem simplificações

e aproximações. Os recursos gráficos das animações podem distanciar do modelo real e dificultar a compreensão de sua totalidade. As distorções ocasionadas da matematização do modelo físico, se não forem explicitadas, resultam em equívocos físicos. É importante destacar que como o modelo computacional é gerado de um modelo físico, ambos possuem um campo de validade. Portanto, na implementação de um software de simulação física, é necessário considerar as restrições e comportamentos do modelo físico. Equívocos na elaboração de softwares de simulações podem comprometer o entendimento da Física. Alguns críticos afirmam que o processo de aprendizagem por erros empíricos é algo que não pode ser “simulado”. (MEDEIROS,2002) A utilização da “Informática no Ensino envolve a análise crítica estabelecida pelas limitações que a modelagem computacional favorece no ensino e suas possibilidades. “Não existe dúvida de que a aprendizagem na interação com seres humanos, com um professor real, é uma necessidade indispensável em muitos casos. Simulações podem ser ricas em imagens e em cálculos; mas são pobres em sua dimensão social.” (MEDEIROS,2002) conclui-se, então, que a simulação computacional verificada na realização deste trabalho não dispensa que seja ministrada por um professor em uma disciplina específica o que será comentado com mais detalhes na seção que comenta sobre “A utilização do software na disciplina de Dinâmica não-linear e Caos”.

2.2 Ensino de Dinâmica Não-Linear e Caos

2.2.1 A utilização do software na disciplina de Dinâmica não-linear e Caos

O software simulador do Oscilador de Duffing em questão foi desenvolvido para posterior utilização na disciplina “Tópicos de Dinâmica Não-Linear e Caos” ofertada pelo “listas de siglas” CPGEI (Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial), da UTFPR, destinada aos estudantes de pós-graduação e extensiva aos estudantes de graduação. Essa disciplina, ministrada pelo professor Mário Sérgio Teixeira de Freitas, do DAFIS “lista de siglas”, tem metodologia diferenciada pois sua estruturação baseia-se em aulas expositivas teóricas, com exemplos de referência e animações e, em aulas práticas, nas quais os alunos desenvolvem simulações computacionais de sistemas não-lineares é incentivado o uso de softwares interativos que demonstrem comportamento não-linear. A disciplina após ser implementada obteve elevado índice de aprovação, ampla aceitação pelos alunos e gerou trabalhos científicos. (FREITAS M. S.

T. 2008)

3 Dinâmica Não-Linear e Caos

Caos é o termo usado para descrever a aparente complexidade do que são considerados sistemas simples. O comportamento caótico parece aleatório, pois é influenciado por fatores externos e por graus de liberdade que influenciam o sistema (HILBORN, 2004).

Os sistemas naturais estudados ***** são determinísticos, pois conhecendo as condições iniciais do sistema, em princípio para dado intervalo, pode-se determinar precisamente as suas condições finais. Mais precisamente, um sistema determinístico é aquele que possui uma regra específica que define o estado do sistema, ou seja, uma função que descreve os pontos do sistema. Porém, conceitualmente, parece-nos conflitante a idéia de um sistema de comportamento aparentemente aleatório e, ao mesmo tempo, determinístico. O que explica esse paradoxo é o conceito de não-linearidade. Se causarmos pequenas perturbações em um sistema não-linear, observam-se grandes divergências na trajetória de fase (KAPLAN, 1995). O estudo dos sistemas caóticos foi possível somente há poucas décadas, pois exige o processamento computacional para a integração de equações diferenciais não-lineares (FREITAS, 2003). Desde 1984, foram incorporados ao vocabulário científico termos como: caos, dinâmica não-linear e fractais. E importantes conceitos da dinâmica não-linear foram desenvolvidos e aplicados em grandes áreas de estudo como: biologia, engenharia, medicina, ecologia, economia e astronomia (KAPLAN, 1995).

O caos pode ser conceituado como um comportamento dinâmico aperiódico em um sistema determinístico cujas alterações nas condições iniciais provocam sensibilidade nos resultados (HILBORN, 2004).

A propriedade aperiódica caracteriza-se por não haver obrigatoriedade na repetição das transições de estado de um sistema. As simulações computacionais dinâmicas e aperiódicas são evidências de caos.

A sensibilidade às condições iniciais é uma característica fundamental para a existência de caos, o que implica que a evolução do sistema pode ser alterada por pequenas perturbações. Isso proporciona maior dificuldade em prever resultados ao longo do tempo em um sistema caótico, pois as condições os alteram significativamente.

Essas propriedades fazem com que o comportamento caótico possua grande

flexibilidade, proporciona aos sistemas intervenções rápidas associadas a determinadas perturbações. Isto é característico de sistemas naturais onde o caos e diversos mecanismos regulatórios fornecem uma flexibilidade incomum (SAVI,2006).

Neste contexto, a descrição ou análise de fenômenos naturais através de modelos ou técnicas não-lineares é mais efetiva ***** do que os modelos ou técnicas lineares. Contudo, as dificuldades inerentes ao estudo dos problemas não-lineares, e o sucesso da mecânica linear em diversas áreas do conhecimento, incentivaram cada vez mais o estudo de modelos lineares e bem comportados.

Ao longo da história, a análise não-linear foi tradicionalmente evitada, criando um paradigma linear. Na citação de Laplace, podemos verificar o determinismo estrito presente no paradigma linear:

“Devemos ver o estado presente do universo como o efeito do seu estado anterior, e como a causa daquele que virá. Uma inteligência que, em qualquer instante dado, soubesse todas as forças pela qual o mundo natural se move e a posição de cada uma de suas partes componentes, e que tivesse também a capacidade de submeter todos estes dados à análise matemática, poderia encompassar na mesma fórmula os movimentos dos maiores objetos do universo e aqueles dos menores átomos; nada seria incerto para ele, e o futuro, assim como o passado, estaria presente diante de seus olhos.” (SAVI,2006).

No final do séc. XIX Henri Poincaré, matemático e físico francês, começou a estudar a estabilidade do Universo. Este estudo viria a se contrapor com o determinismo estrito de Laplace, como pode-se ver na citação de Poincaré: “Uma causa muito diminuta, que nos escapa, determina um efeito considerável, que não podemos deixar de ver, e então dizemos que este efeito é devido ao acaso. Se pudéssemos conhecer exatamente as leis da natureza e a situação deste mesmo universo no instante subsequente, mesmo quando as leis naturais já não tivessem mais segredo para nós, só poderíamos conhecer a situação inicial aproximadamente. Se isto nos permite aproximar a situação subsequente com o mesmo grau de aproximação, ficamos satisfeitos, dizemos que o fenômeno foi previsto, que é governado por leis. Mas nem sempre isto ocorre; pode acontecer que diferenças mínimas nas condições iniciais produzam diferenças muito grandes no fenômeno final; um erro mínimo nas primeiras produziria um erro enorme nesses últimos. A previsão torna-se impossível e temos o fenômeno ao acaso.” (SAVI,2006)

Apesar de Poincaré ter uma visão clara sobre a sensibilidade do sistema às

condições iniciais, foi só em 1963 com Lorenz, que houve uma análise mais aprofundada de sistemas dinâmicos. Ele desenvolveu estudos sobre problemas atmosféricos. Contando com o auxílio de um computador, Lorenz tratava o modelo de Rayleigh-Benard para a convecção de fluidos, e observou que uma pequena variação nas condições iniciais poderia acarretar grandes diferenças na evolução do sistema (SAVI,2006). O comportamento caótico não-linear do oscilador de Duffing pode ser observado com os retratos de fase do movimento que são gráficos em que, no eixo das ordenadas, se encontram os valores de velocidade e nas abcissas, valores de posição. Os parâmetros para a análise desse movimento são: o coeficiente de amortecimento viscoso do meio, a constante de elasticidade da haste metálica, a atração magnética dos ímãs, a frequência e a amplitude da oscilação de fonte externa****REFERENCIA AO HOLMES SOBRE OS PARAMETROS****. Inicialmente, o usuário fornece os dados iniciais da velocidade e da posição da partícula. O sistema modelado matematicamente com a utilização de equações diferenciais. E sua resolução envolve métodos de integração numéricos. O estudo do sistema de Duffing não é de grande complexidade, pois envolve osciladores que podem ser tratados unidimensionalmente e as estruturas fractais são identificadas com a "seção de Poincaré". A "seção de Poincaré" é um modo de detectar soluções periódicas para a observação de um corte transversal (FREITAS, 2003). Denomina-se também "amostragem estroboscópica" nos sistemas dependentes do tempo. A simulação numérica e a interpretação geométrica do plano de fase são recursos que podem ser utilizadas desde a sistemas simples, como o oscilador harmônico simples, a sistemas complexos, como osciladores com excitação externa e amortecimento viscoso.(FREITAS, 2003)

3.1 Oscilador de Duffing



Figura 1: *O Oscilador de Duffing*

A equação diferencial que caracteriza o oscilador de Duffing pode ser descrita como:

$$\frac{d^2x}{dt^2} = -x^3 + \alpha x - 2\beta \frac{dx}{dt} + A \cos(\omega t) \quad (1)$$

A Figura ?? ilustra o Oscilador de Duffing, que consiste em uma haste metálica de elasticidade α fixada a um suporte que oscila com amplitude A e frequência angular ω , tal oscilação realiza o papel de uma força excitadora externa. O oscilador está sujeito, além da força mencionada, ao amortecimento viscoso β do meio em que está inserido e à ação de dois ímãs idênticos equidistantes da posição da haste. É ilustrado, também, o ponto estudado que se move segundo um arco de circunferência, porém é adotado que este se move segundo um eixo x em uma direção horizontal. Esta aproximação pode ser feita pois a amplitude de oscilação deste ponto é pequena comparada ao tamanho da haste.

A função da energia potencial deste oscilador é descrita como:

$$U(x) = -\frac{1}{2}kx^2 + \frac{1}{4}x^4 \quad (2)$$

Baseado em experiências com o oscilador harmônico simples, sabe-se que a partícula estudada reagirá com um movimento de amplitude relativamente grande quando a frequência da força externa coincide com a frequência de oscilação natural da mesma partícula. O problema é que a frequência de oscilação natural depende da amplitude deste movimento. Então como a partícula começa a reagir à força excitadora externa, a amplitude muda e por isso, a frequência de oscilação natural também muda. Várias características singulares podem aparecer (HILBORN, 2004, p. 581):

1. A curva de resposta do sistema muda com o aumento da amplitude da força excitadora externa;
2. The response curve of the system shows hysteresis: the response amplitude depends on whether we increase the frequency through the resonance region or decrease the frequency through the resonance region.
3. O sistema pode apresentar comportamento caótico.

3.1.1 Gráficos

- Plano de Fase: O plano de fase é um gráfico onde, no eixo das abcissas, constam os valores da posição da ponta da haste do oscilador e, o eixo das ordenadas, constam os valores da velocidade deste mesmo ponto.
- Seções Temporais: As seções temporais são gráficos onde o eixo das abcissas corresponde ao tempo e o eixo das ordenadas é o dado que se pretende analisar por exemplo: Gráfico da posição contra o tempo, velocidade contra o tempo e da aceleração contra o tempo.

- Seção de Poincaré: A seção de Poincaré, é obtida através de uma análise estroboscópica no plano de fase, ou seja, registramos a posição da ponta da haste do oscilador em um dado instante inicial e repetimos isso periodicamente em um certo intervalo de tempo adotado. Veremos que as posições nesses instantes formam o que é chamado de Seção de Poincaré.
- Gráfico de Energia: O esboço do gráfico da energia, pode ser obtido pela plotagem da equação (2) e o gráfico desta função é chamado de double-well. O nome "double-well" está relacionado à correspondência entre a função da energia potencial com a estrutura da mola (HILBORN,2004).



Figura 2: *Plano de Fase*

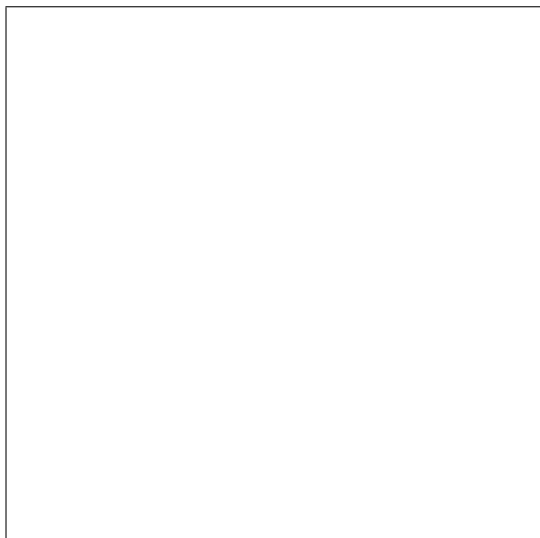


Figura 3: *Seção de Poincaré*



Figura 4: *Gráfico de Energia*

4 Software

Segundo PRESSMAN, 1995 software pode ser caracterizado como: "(1) instruções (programas de computador) que, quando executados, produzem a função e o desempenho desejados; (2) estruturas de dados que possibilitam que os programas manipulem adequadamente a informação; e (3) documentos que descrevem a operação e o uso dos programas". Ainda citando o mesmo autor, ele também define três características que distinguem softwares de hardwares:

1. O software é desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico;
2. Software não se "desgasta";
3. A maioria dos softwares é feita sob medida em vez de ser montada a partir de componentes existentes.

Pressman divide os softwares em sete grupos diferentes: Básico, de Tempo Real, Comercial, Científico e de Engenharia, Embutido, de Computação Pessoal, de Inteligência Artificial.

O software a ser desenvolvido neste projeto encaixa-se no grupo Científico e de Engenharia. Este grupo tem sido caracterizado por algoritmos de processamento de

números. Suas aplicações variam desde astronomia até vulcanologia.

4.0.2 Engenharia de Software

Engenharia de Software é o estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais (PRESSMAN, 1995). Além do aspecto econômico esta disciplina através de seus métodos e ferramentas possibilita um desenvolvimento racional e oportuno do software de computador (PRESSMAN, 1995).

4.1 Análise de Aplicativos Similares

A Figura 7 apresenta uma análise que tem como objetivo a busca e o levantamento dos pontos fortes e fracos de aplicativos similares ao proposto neste projeto. Este tipo de estudo permite que o projeto tenha tanto uma base para seu início quanto uma base de comparação. O estudo foi tabulado, sendo as colunas representando os aplicativos e as linhas os elementos avaliados. A variável de avaliação tem três estados possíveis como se pode ver na Figura 6.



Figura 5: *Legenda da Análise*

A classificação positivo é dada quando a ferramenta avaliada existe no aplicativo e sua configuração é satisfatória. Já negativo é atribuída quando a ferramenta existe mas sua configuração é confusa ao ponto de poder ser considerada não-satisfatória. E por último a classificação não possui quando o aplicativo não disponibiliza esta ferramenta. A seguir os endereços eletrônicos onde foram encontrados os aplicativos em questão e a tabela comparativa.

- A - <http://www.math.udel.edu/~hsiao/m302/JavaTools/osduffng.html>
- B - <http://www.maths.usyd.edu.au/u/nigel/NLD/Duffing/Duffing.html>
- C - <http://www.compadre.org/OSP/document/ServeFile.cfm?ID=7857&DocID=642>
- D - <http://www.df.ufcg.edu.br/~adriano/ClickApp2.html>
- E - <http://brain.cc.kogakuin.ac.jp/~kanamaru/Chaos/e/Duffing2/>
- F - Software desenvolvido neste trabalho.

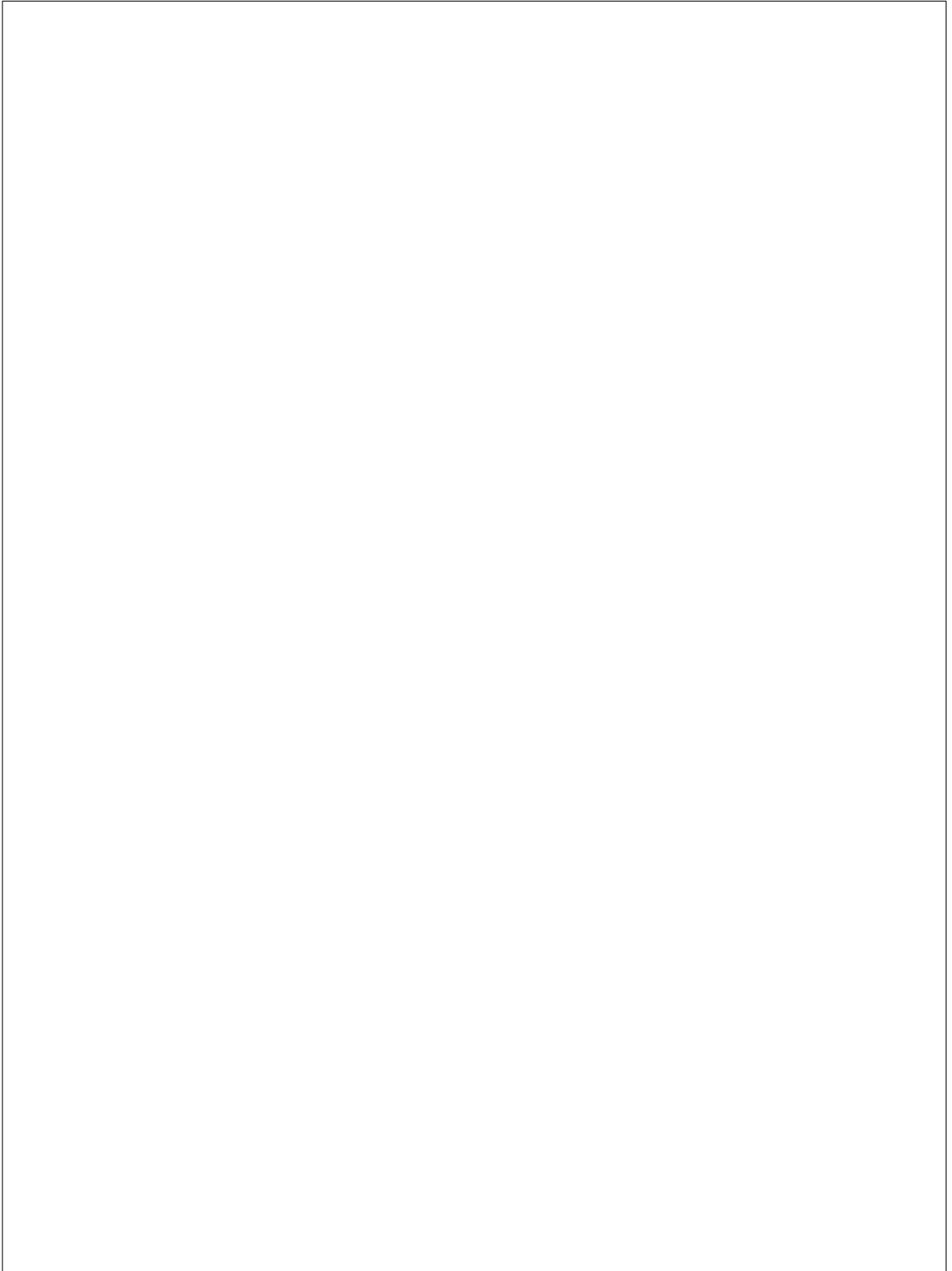


Figura 6: *Análise*

4.2 Padrões de Projeto

Este trabalho utiliza o que é chamado de Padrões de Projeto ou Design Patterns. Estes padrões são ferramentas que permitem um melhor desenvolvimento do aplicativo, pois eles solucionam muitos dos problemas que os projetistas enfrentam no dia a dia (Gamma et al,2000 p.27).Outro benefício que estas ferramentas trazem está na possibilidade de reutilização de soluções, algo que pode acelerar o desenvolvimento do aplicativo.

4.2.1 Padrão MVC

A sigla MVC vem do inglês (Model, View, Controller) que pode ser traduzido como Modelo, Visão e Controle. Segundo Lopes,2006 MVC é um padrão de arquitetura de Software criado na linguagem Smalltalk e tem como objetivo desacoplar a interface gráfica do código que realizará o trabalho. Abaixo uma breve descrição de como atua cada parte desta arquitetura:

- Visão: tem como função mostrar dados e receber informações adicionadas pelo usuário. Controle: faz a comunicação entre os dados recebidos pela interface e o modelo. Também nesta camada ocorre o processamento da informação que sera enviada ao modelo.
- Modelo: é a parte do software que irá processar dados e que executará as funções, cálculos, métodos necessários para cumprir o objetivo do programa.

Um dos padrões de projetos utilizados neste tipo de abordagem é o padrão observer (observador) que é definido por Gamma como o padrão que “..define uma dependência um-para-muitos entre objetos para que quando um objeto mudar de estado, todos os seus dependentes sejam notificados e atualizados automaticamente.”Este padrão é muito importante neste projeto, uma vez que várias das ferramentas do software irão mostrar informações provenientes de uma mesma fonte, mas de formas diferentes. Além disso o uso desse padrão permite uma expansão do sistema de forma sustentável, pois se for preciso adicionar uma nova ferramenta, não será necessário mexer no cerne do programa. (FERNANDES,2006)

4.2.2 Importância do Padrão de Projeto Observer no Sistema

O padrão observer tem um grande destaque neste trabalho, pois o seu funcionamento é uma das peças cruciais para que o sistema possa propiciar a experiência objetivada ao usuário. Isto ocorre pois este padrão permite que vários objetos sejam atualizados automaticamente de acordo com mudanças no modelo. Sua principal aplicação está na parte de visualização de dados em que são necessárias várias representações diferentes dos mesmos, o que é o caso deste trabalho. Este padrão tem como dois atores uma interface chamada Observador e uma classe abstrata Observado. A classe Observado conhece todas as classes que implementam a interface Observador. Assim quando o Observado é atualizado, ele notifica todos os observadores de sua mudança. Estes por sua vez recebem os novos dados e os tratam da maneira que precisam para executar sua função. No caso deste projeto todas as ferramentas implementam a interface Observador e o modelo deriva da classe Observado, o que significa que quando houver uma atualização no modelo, todos as ferramentas serão atualizadas.

Exemplo de funcionamento:

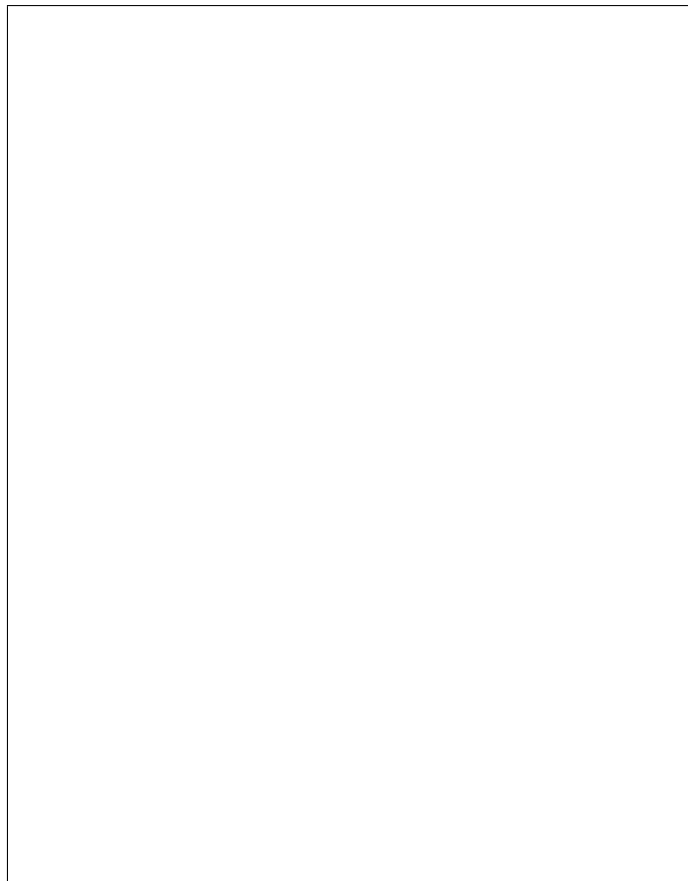


Figura 7: *Exemplo de Funcionamento do Observer*

O método `setDados` da classe `Modelo` adiciona o novo tempo a função de integração fazendo com que o modelo tenha os próximos valores do sistema. Além disso ele chama o método `notificar` herdado da super classe abstrata `Observado`. Esse método por sua vez chama em todos os observadores o método `update`. Este método tem sua implementação específica em cada ferramenta juntamente para que cada um trate os dados da maneira conveniente. O código acima mostra o exemplo dessa comunicação para a classe da `JanelaPlanoDeFase` e `JanelaMostrador`.

4.3 UML

A Unified Modeling Language é atualmente o esquema de representação gráfica mais amplamente utilizado para modelar sistemas orientados a objetos.(DEITEL e DEITEL, 2005).

4.4 JAVA

A linguagem Java foi escolhida para o desenvolvimento deste projeto por ser uma linguagem geralmente utilizada com o paradigma de orientação a objetos, além de sua grande popularidade. Esta linguagem tem mais de 6.5 milhões de desenvolvedores . A popularidade da linguagem assegura um grupo imenso de recursos de suporte (BARNES; KOLLING, 2004). E quanto mais recursos mais fáceis será superar os obstáculos que podem vir a ser encontrados durante a fase de desenvolvimento do aplicativo. Além destes motivos também foi levado em conta que é a linguagem de programação comum a todos os integrantes do projeto, assim aproveitando ao máximo os conhecimentos já adquiridos pelos componentes da equipe. Outro ponto a favor desta linguagem é sua vasta API (application programmng interface), que possui vários conjuntos de classes prontas que o desenvolvedor pode utilizar-se.

4.5 Interface Gráfica

O sistema desenvolvido possui uma grande gama de ferramentas, sendo o usuário a decidir quais são relevantes no momento em que está interagindo com o programa. Para propiciar essa liberdade, optou-se por uma abordagem com janelas internas ao programa, sendo elas facilmente fechadas e abertas. O projeto das janelas deste pro-

grama difere um pouco de outros, uma vez que normalmente este tipo de janela pode ser fechado, minimizado e aberto. Neste caso, como as janelas são predefinidas elas não podem ser minimizadas, pois não existe uma função relevante para tal tipo de comportamento. Assim o comportamento básico das janelas é aberto ou fechado dentro do programa. As janelas ativas ficam em uma área grande do aplicativo e existe uma outra área menor onde ficam os ícones que representam as janelas que não estão em uso. Essa divisão existe para que o usuário tenha um acesso rápido a todas as ferramentas que desejar, pois caso clicar no ícone referente a ferramenta desejada, esta irá abrir para ser utilizada. A maioria das ferramentas são gráficos que auxiliam no estudo do movimento caótico executado pelo sistema.

4.5.1 Ícones

Os ícones das ferramentas possuem cada um uma cor, uma imagem representativa e o nome da ferramenta. Esses três tipos de informação tem como objetivo diferenciar uma ferramenta da outra, tentando evitar confusão.

4.6 Análise Orientada a Objetos

4.6.1 Descrição do Problema

Para que o software desenvolvido atenda aos objetivos iniciais deste projeto, deve propiciar a todos seus usuários uma experiência satisfatória. Os atores envolvidos são basicamente de dois tipos podendo haver desdobramentos, mas estes não devem fugir muito dos perfis iniciais. São eles:

- **Professores** Este tipo de usuário deve usar o sistema como uma ferramenta de ensino ou para alguma atividade similar. Ele deve ter acesso a todas as funcionalidades e o programa deve atender a necessidade explanativa que uma aula possa vir exigir.
- **Alunos/Interessados** Este ator difere do anterior em alguns fatores. Neste grupo existe a possibilidade de que o usuário não seja um profundo conhecedor do tema, e essa limitação não deve o afastar do produto deste projeto. Outro fator importante é o caráter explorativo que o software pode propiciar, fazendo com que para aqueles que não tem conhecimento nenhum, ao menos possam se familiarizar com elementos do objeto de estudo.

4.6.2 Requisitos Funcionais do Sistema

As funcionalidades de software são as mesmas para os dois atores.

Elementos para o estudo do sistema conceitual:

- Gráfico da posição contra o tempo;
- Gráfico da velocidade contra o tempo;
- Gráfico da aceleração contra o tempo;
- Gráfico da velocidade contra a posição (Plano de Fase);
- Gráfico de Energia do Sistema;
- Controles para interação com o usuário;
- Controle da constante elástica da barra;
- Controle da constante de amortecimento do meio;
- Controle da amplitude da força excitadora;
- Controle da frequência angular da força excitadora;
- Controle da velocidade inicial do movimento;
- Controle da posição inicial do movimento;
- Pausar a simulação;
- Entrada de dados numericamente (por input box);

4.6.3 Diagrama de Classes



Figura 8: *Diagrama de Classes*

4.6.4 Diagramas de Sequencia

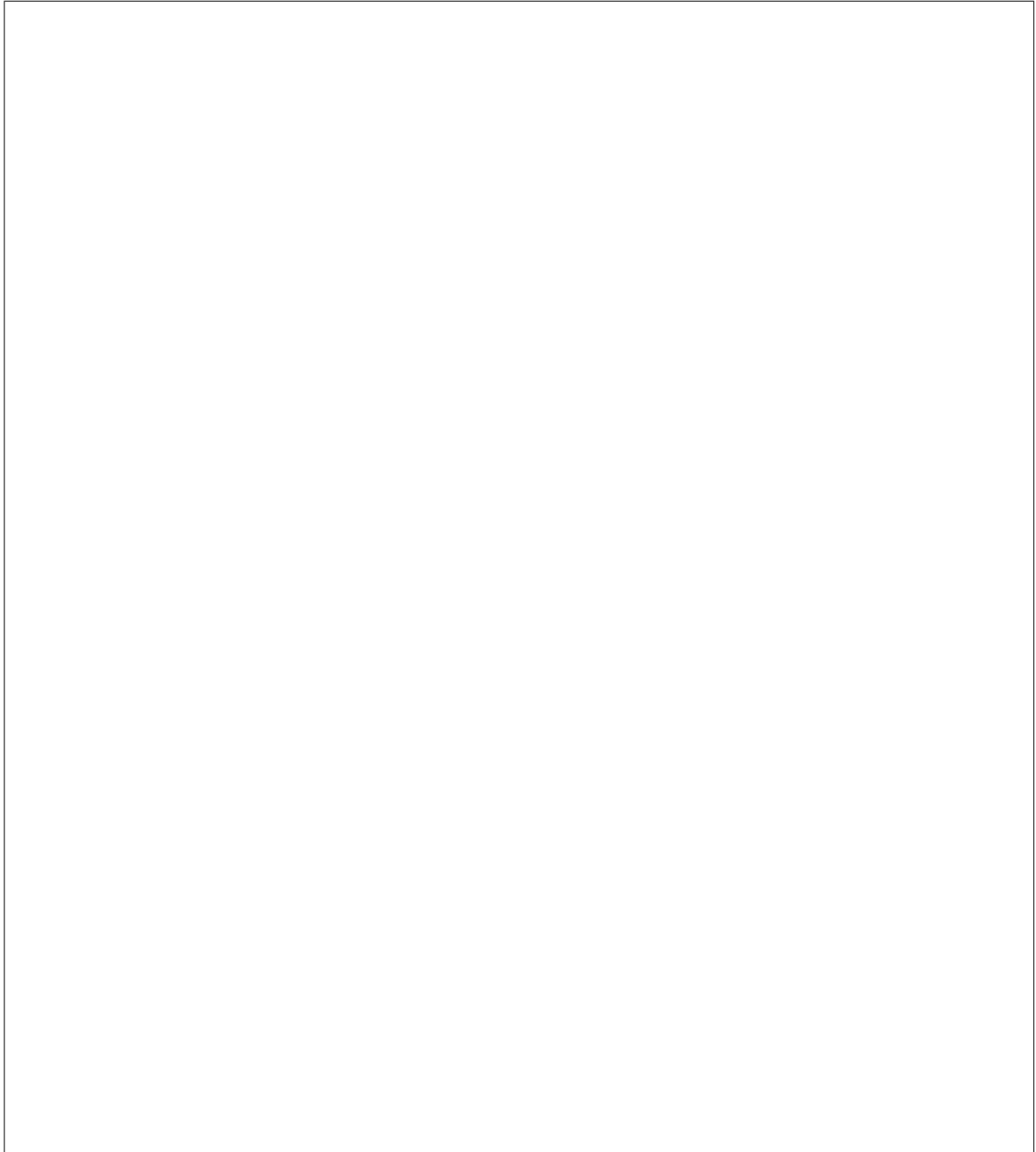


Figura 9: *Diagrama de Entrada de Dados*

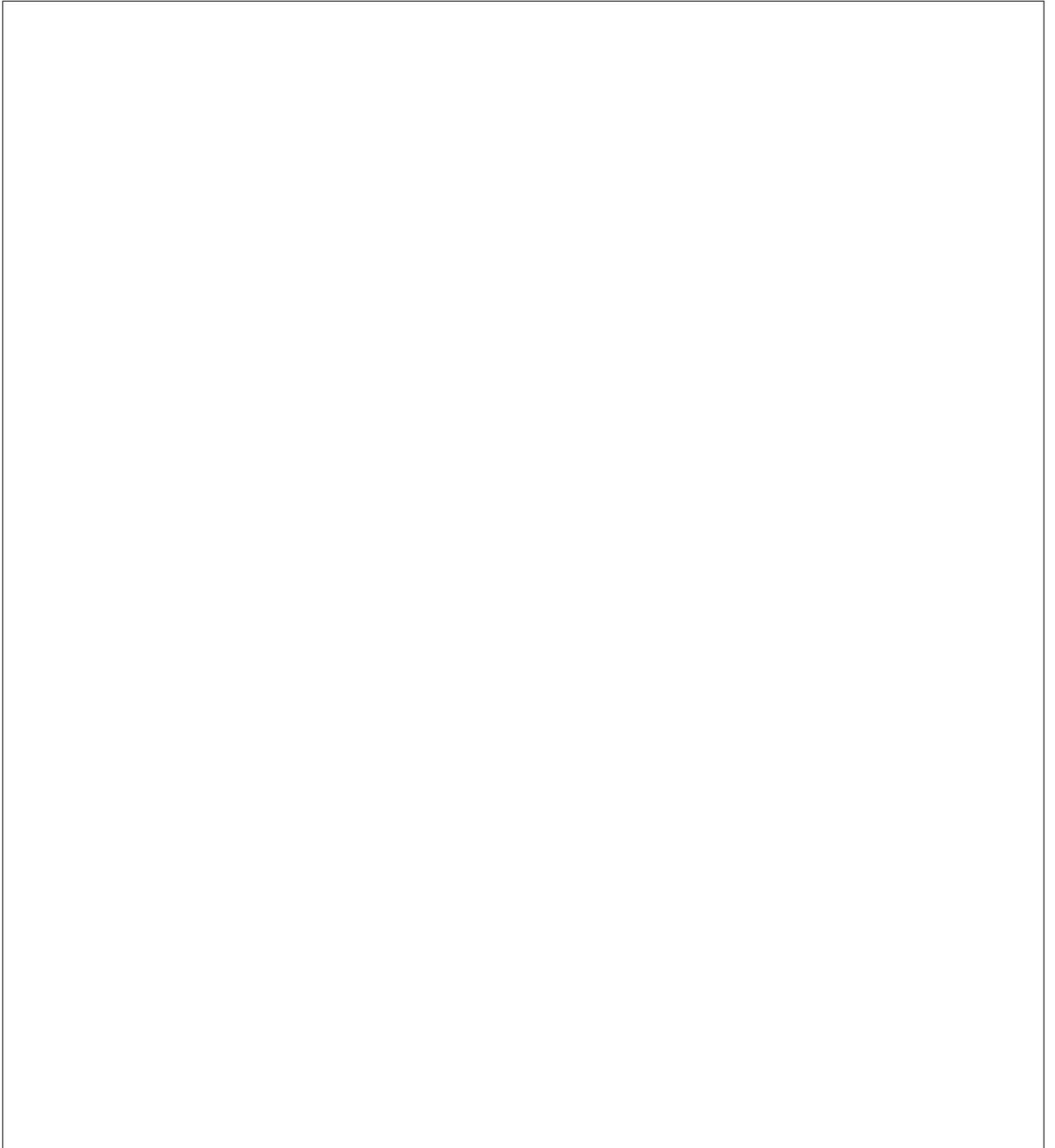


Figura 10: *Diagrama de Ações do Usuário*

4.6.5 Dicionário de Classes

- Controle

Classe que reuni os comandos que o usuário pode executar na interface gráfica, e os repassa para o modelo.

- Modelo

Classe que executa a integração da equação diferencial deste estudo, armazenando e devolvendo os valores a serem mostrados pelas ferramentas do programa.

- Observado

Classe abstrata com métodos de comunicação com objetos que implementem a interface observador.

- Observador

Interface que cria a relação com a classe abstrata Observado. Essa relação tem com intuito a atualização das ferramentas sendo usadas no sistema.

- PrincipalJanela

Classe que cria a janela principal do aplicativo. Derivada da classe JFrame.

- DesktopPrincipal

Classe responsável pela área em que as ferramentas ativas são mostradas. Derivada da classe JDesktopPane

- PainelInferior

Classe responsável pela área de janelas desativadas. Derivada da classe JPanel.

- BotaoInter

Classe que cria o botão que representa as janelas desativadas. Derivada da classe JButton

- JanelaInterna

Super classe de todas as janelas internas. Nela estão os comportamentos comuns a todas as ferramentas. Derivada da classe JInternalFrame e implementa a interface Observador

- JanelaConfig

Classe que mostra a configuração física do objeto simulado. Derivada da classe JanelaInterna.

- JanelaEnergia

Classe que representa a energia do sistema. Derivada da classe JanelaInterna.

- PainelEnergia

Classe que desenha o gráfico da energia do sistema. Derivada da classe JPanel.

- JanelaEntradaDados

Classe que em que pode-se inserir valores para alterar o comportamento do sistema. Derivada da classe JanelaInterna.

- PainelEntradaDados

Classe que desenha botões, campos, legendas referentes a ferramenta de entrada de dados. Derivada da classe JPanel.

- JanelaGrafico

Super classe dos gráficos com uma variável contra o tempo. Possui todo o comportamento padrão deste tipo de gráfico no sistema. Derivada da classe JanelaInterna.

- PainelGrafico

Classe que desenha o gráfico da classe JanelaGrafico. Derivada da classe JPanel.

- JanelaXT

Classe que representa a ferramenta referente ao gráfico do espaço pelo tempo. Derivada da classe PainelGrafico.

- JanelaVT

Classe que representa a ferramenta referente ao gráfico da velocidade pelo tempo. Derivada da classe PainelGrafico.

- JanelaAT

Classe que representa a ferramenta referente ao gráfico do espaço pelo tempo. Derivada da classe PainelGrafico.

- JanelaMostrador

Classe que apenas mostra os valores obtidos pelo modelo. Derivada da classe JanelaInterna.

- PainelMostrador

Classe que desenha na tela os valores referentes a classe JanelaMostrador. Derivada da classe JPanel.

- JanelaPlanoDeFase

Classe que representa a ferramenta referente ao gráfico da velocidade pelo espaço. Derivada da classe PainelGrafico.

- PainelPlanoDeFase

Classe que desenha o gráfico da classe JanelaPlanoDeFase. Derivada da classe JPanel.

- JanelaPoincare

Classe que representa a ferramenta referente ao gráfico da velocidade pelo espaço. Derivada da classe PainelGrafico.

- PainelPoincare

Classe que desenha o gráfico da classe JanelaPoicare. Derivada da classe JPanel.

5 Considerações Finais

5.1 Discussão

Como resultado desta pesquisa temos um protótipo do aplicativo proposto inicialmente. Esse protótipo possui todos os requisitos funcionais levantados no começo do projeto exceto o esboço do gráfico da seção de Poincaré efetivamente pois por alguma incompatibilidade de natureza numérica o corte de Poincaré é plotado em meio a uma poeira de pontos o que dificulta a visualização, mas não impossibilita que tal etapa seja atingida após vários testes afim de encontrar uma configuração mais adequada. Testes com usuários se fazem necessários para validar a eficiência da abordagem escolhida. Assim

um próximo passo para este projeto seriam estas avaliações (testes informais foram feitos durante a produção do aplicativo, mas não documentados) e mais importante ainda, testes em salas de aula. Avaliações em sala de aula seriam de grande interesse pois observação em campo pode levantar inúmeras situações não previstas pelos envolvidos no projeto, o que ajudaria ainda mais na evolução deste software. A versão protótipo do Simulador Interativo do Oscilador de Duffing foi executada em versão para desktop. Outro possível aperfeiçoamento seria uma versão online do software. Porém este tipo de adaptação exigiria um outro tipo de conhecimento técnico e teórico, pois a experiência que o hipertexto proporciona é diferente da proporcionada por um aplicativo offline.

5.2 Resultados

O software desenvolvido é composto por nove janelas internas (como se pode ver na Figura 12), que em conjunto simulam o comportamento caótico não-linear do Oscilador de Duffing. A janela interna de Entrada de Dados permite a inserção de parâmetros adicionais no estudo desse movimento, a barra deslizante proporciona ao usuário o controle de velocidade de simulação do software. O aplicativo possibilita a visualização da sensibilidade às condições iniciais do oscilador. O Espaço de Configuração é uma animação da ponta haste metálica se deslocando em uma dimensão. O gráfico de Energia Potencial apresenta comportamento regular, pois a equação de Energia Potencial está em função unicamente da posição e esta varia dentro de um intervalo definido de valores. As seções temporais, posição por tempo, velocidade por tempo e aceleração por tempo apresentam comportamento não-periódico dependendo dos valores iniciais e dos parâmetros que são inseridos. O software proporciona um redimensionamento da maioria das janelas internas para melhor visualização e possui botões de abertura e fechamento das mesmas. O Plano de Fase refere-se à plotagem da velocidade pela posição. A força excitadora do oscilador mantém frequência constante, a cada iteração proveniente de um período completo desta força é plotado, em um plano de fase, a posição e a velocidade neste instante. O resultado dessa iteração dá origem a Seção de Poincaré.



Figura 11: *Programa com as Janelas Internas Abertas*

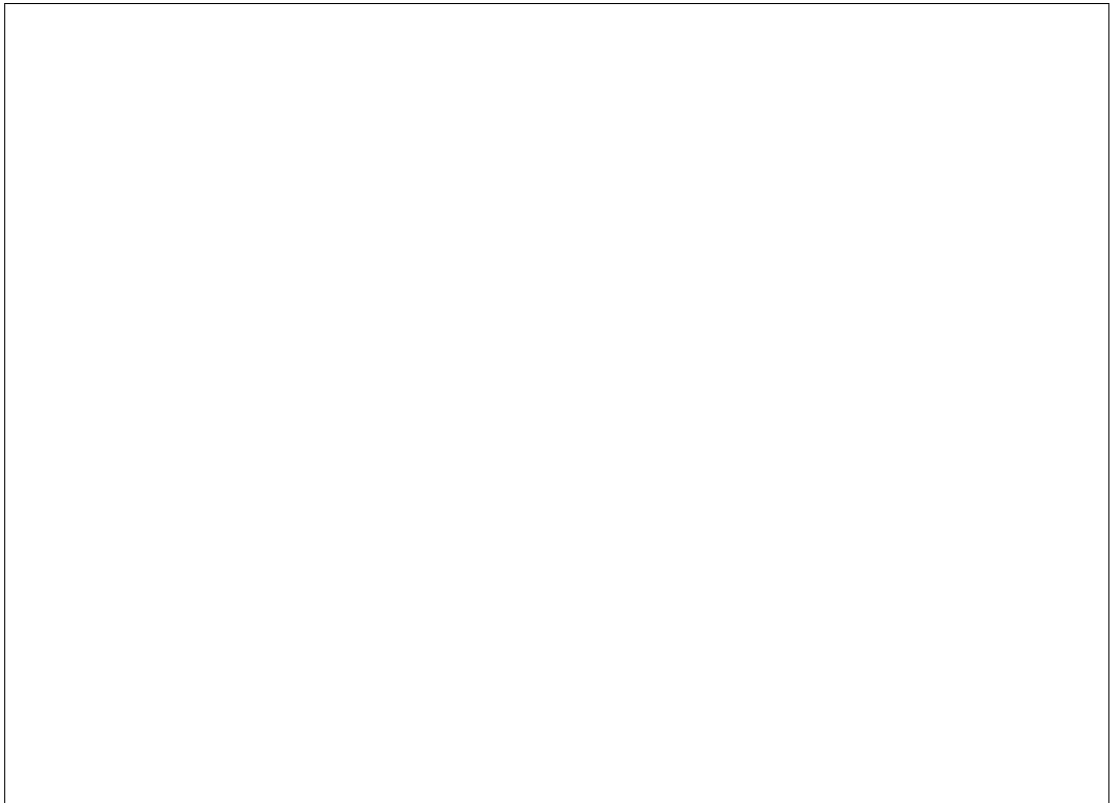


Figura 12: *Programa com as Janelas Internas Fechadas*

5.3 Conclusão

O trabalho desenvolvido baseado nas propostas anteriormente apresentadas atingiu grande parte do objetivo proposto pois podemos simular o oscilador não-linear bi-estável caótico de Duffing e, por conseguinte, identificamos o seu comportamento caótico, por exemplo, através de sua sensibilidade às condições iniciais. Nesse contexto, a criação de um simulador do Oscilador de Duffing contribui para a aplicação de conceitos de não-linearidade e caos. É relevante que o estudante de física, mesmo que com recursos limitados, possa correlacionar aspectos teóricos com práticos (GLEISER, 2000).

6 Referências Bibliográficas

1. BARNES, David; KOLLING, Michael. Programação Orientada a Objetos com JAVA. São Paulo: Pearson Prentice Hall, 2004.
2. BAXTER, Mike. Projeto de Produto. Guia prático para o design de novos produtos. São Paulo: Editora Edgard Bluncher, 2005.
3. BOLDO, E. M; CIANI, A. B. Modelagem Computacional De Osciladores Não-Lineares Para a Aprendizagem de Equações Diferenciais, 2007.
4. BONATTO, Cristian. Estrutura de Diagramas de Fase de Sistemas Dinâmicos de Tempo Contínuo, 2008. Tese de Doutorado
5. BRANDÃO, R.V; ARAÚJO, I.S; VEIT, E.A, Modelagem e ensino de física, Física na Escola, Vol. 9, No. 1, 2008
6. CHAVES, Eduardo. Tecnologia: Um Novo Paradigma na Escola. Disponível em http://www.microsoft.com/brasil/educacao/parceiro/novo_paradi.msp acesso em 10 de Março de 2009.
7. COELHO NETTO, José Teixeira. Semiótica, Informação e Comunicação. São Paulo: Perspectiva, 1989.
8. CUNHA, Maria C. Métodos Numéricos. 2ª Ed. Campinas, São Paulo: Editora da UNICAMP, 2000.
9. DEITEL, Harvey; DEITEL, Paul J. Java Como Programar Sexta Edição. São Paulo: Pearson Prentice Hall, 2005.
10. FERNANDES, Rafael Lopes. Padrão MVC com Java Swing. Maranhão, 2006. Disponível em: http://www.dai.cefet-ma.br/rafaelf/ensino/2006.2/lp3/arquivos/swing_mvc.pdf. Acesso em: 23 de Abril de 2009.
11. FLORES-ANDO, Fortunato; AGUILAR-IBÁÑEZ, Carlos; SUÁREZ-CASTAÑÓN, Miguel S. An Identificaion Genetic Algorithm for a Family of Duffing's System.
12. FREITAS, Mário Sérgio Teixeira. Introdução ao Caos: Contribuição metodológica do DAFIS para a pós-graduação em engenharia. Revista Tecnologia & Humanismo UTFPR, Curitiba, n. 35, Segundo semestre, 2008.

13. FONTOURA, Antônio Martiniano. *Edade A Educação de Crianças e Jovens Através do Design*. Florianópolis, 2002. Tese de Doutorado. UFSC.
14. GAMMA, Erich; HELM, Richard; Johnson, Ralph; VLISSIDES, John. *Design Patterns: Elements of Reusable Objectoriented Software*. Addison-Wesley Longman, 1995.
15. GARRET, J.J. *The Elements of User Experience*. New York: New Riders, 2003.
16. GLEISER, Marcelo. Por quê ensinar Física. Artigo publicado pela Revista Física na Escola, Vol.1, No. 1, 2000.
17. HILBORN, Robert C. *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*. Second Edition, Oxford University Press, 2000.
18. HOLMES, Phil. *A Nonlinear Oscillator with a Strange Attractor*. Ithaca, New York: Cornell University, 1978. Artigo publicado por The Royal Society A: Mathematical, Physical and Engineering Sciences.
19. KAPLAN, Daniel; GLASS Leon. *Understanding Nonlinear Dynamics*. New York: Springer-Verlag, 1995.
20. MEDEIROS, Alexandre. MEDEIROS Cleide Farias de. Possibilidades e Limitações das Simulações Computacionais no Ensino da Física Revista Brasileira de Ensino de Física, v. 24, n. 2, Junho, 2002.
21. MERKLE, Luiz Ernesto; Arandi G. Bezerra, Jr; GIMÉNEZ-LUGO, Gustavo Alberto; *Interdisciplinariedade e Críticidade na Formação em Engenharia de Computação na UTFPR, Tecnologia e Humanismo*, 2008.
22. MOURA, Mônica. *Design de Hipermedia: Novo Campo de Ação No Ensino, Na Aprendizagem e na Formação Profissional*. In: Congresso Nacional de Ambientes Hipermedia para Aprendizagem. Florianópolis, 2004.
23. NILSEN, Jakob. *Usability Engineering*. Academic Press, 1993.
24. PERUZZI, Nelson José. *Dinâmica Não-Linear e Controle de Sistemas Ideais e Não-Ideais Periódicos*. Campinas, 2005. Tese de Doutorado. UNICAMP.

25. PRESSMAN, Roger S. Engenharia de Software. Pearson Makrin Books, 1995.
26. ROMISZOWSKI, Hermelina Pastor. Avaliação no Design e Desenvolvimento de Multimídia Educativa: Estratégia de Apoio ou Parte do Processo? In.: Anais do VII Congresso Internacional de Educacao a Distância da Associação Brasileira de Educação a Distância - ABED, São Paulo, 2000. Disponível em <http://www.abed.org.br/publique/cgi/cgilua.exe/sys/start.htm?UserActiveTemplate=4abed&infoid=184&sid=102&tpl=printerview> acesso em 21 de Março de 2009.
27. SAVI, Marcelo A. Dinâmica Não-Linear e Caos. Rio de Janeiro: Editora E-Papers. 2006.
28. Teixeira Coelho Netto, J. Semiótica, Informação e Comunicação. Editora Perspectiva, 1999.
29. UTFPR(2006). Projeto Pedagógico do Curso de Engenharia de Computação.
30. VIEIRA, Josiane Wanderlinde. O Ensino da Geometria Descritiva para Alunos Surdos Apoiado em um Ambiente Hipermídia de Aprendizagem. Florianópolis: Visual gd, 2005.
31. VIET, E.A; MORS, P.M; TEODORO, V.D. Ilustrando a Segunda Lei de Newton no Século XXI, Revista Brasileira de Ensino de Física, Vol. 24, No. 2, Junho, 2002.

7 Apêndice

7.1 Um breve manual de utilização

7.1.1 Janelas Internas

- Gráfico $X \ t$

Apresenta o gráfico da seção temporal em que contém valores de posição no eixo das ordenadas e valores de tempo no eixo das abcissas;

- Gráfico $V \ t$

Apresenta o gráfico da seção temporal em que contém valores de velocidade no eixo das ordenadas e valores de tempo no eixo das abcissas;

- Gráfico $A \ t$

Apresenta o gráfico da seção temporal em que contém valores de aceleração no eixo das ordenadas e valores de tempo no eixo das abcissas;

- Gráfico $E \ x$

Apresenta o gráfico em que contém valores de energia potencial no eixo das ordenadas e valores de posição no eixo das abcissas;

- Dados Analíticos

Apresenta o valor instantâneo de posição, velocidade e aceleração;

- Plano de Fase

Apresenta o Plano de Fase do oscilador, ou seja, o gráfico que apresenta valores de velocidade no eixo das ordenadas e valores de posição no eixo das abcissas;

- Seção de Poincaré

Apresenta, em um plano de fase, um protótipo da Seção de Poincaré;

- Entrada de Dados

Abrirá uma janela interna que irá conter:

– Pos Inicial

Neste campo, o usuário pode configurar a posição inicial da ponta da haste metálica do oscilador. Para resultados satisfatórios, é recomendado que o usuário coloque valores no intervalo de -2.0 a 2.0.

– Vel Inicial

O usuário pode configurar a velocidade inicial da ponta da haste metálica do oscilador. Para resultados satisfatórios, é recomendado que o usuário coloque valores no intervalo de -2.0 a 2.0.

– Coef Elast

Neste campo o usuário altera o coeficiente elástico da haste metálica.

– Amort Visc

Neste, o usuário pode configurar o amortecimento viscoso do meio em que o oscilador está inserido.

– Amplitude Fexc

O usuário configura, neste campo, a amplitude de movimento que a força harmônica excitadora desempenha.

– Freq Fexc

Neste campo, é configurado a frequência da força harmônica excitadora.

– Botão Iniciar

Inicia simulação.

– Botão Pausar

Pausa simulação.

– Barra deslizante inferior

Controla a velocidade de simulação. Vale a pena ressaltar que esta velocidade depende da máquina em que o programa está sendo executado e que a quantidade de janelas interiores abertas também influem neste desempenho.

● Espaço de Configuração

Abrirá uma janela que irá conter a ponta da haste metálica representada por um círculo vermelho se movendo na direção do eixo das abcissas. Conterá, também, dois quadrados azuis que representam a posição de cada ímã.

7.1.2 Abrindo Janelas Internas

O usuário pode escolher a janela interna que gostaria de visualizar. Para isto basta posicionar o ponteiro do mouse em algum dos botões localizados na parte inferior da janela principal (Figura 13) e verificar, logo acima, o texto que este contém e, com isso, selecionar a janela desejado.

7.1.3 Iniciando Simulação

Com o aplicativo aberto, assim como na Figura 13 o usuário poderá clicar, com o ponteiro do mouse, encima do penúltimo botão localizado na parte inferior da janela principal no sentido da esquerda para a direita. Assim abrirá a janela interna Entrada de Dados onde o usuário poderá iniciar a simulação no botão Iniciar bem como pausar quando quiser clicando no botão Pausar.